MATLAB code used in Kobayashi & Matsuo, *Cell Reports* (2023)

1. License
2. Generate Demo Data
3. Calcium Event Detetion

License

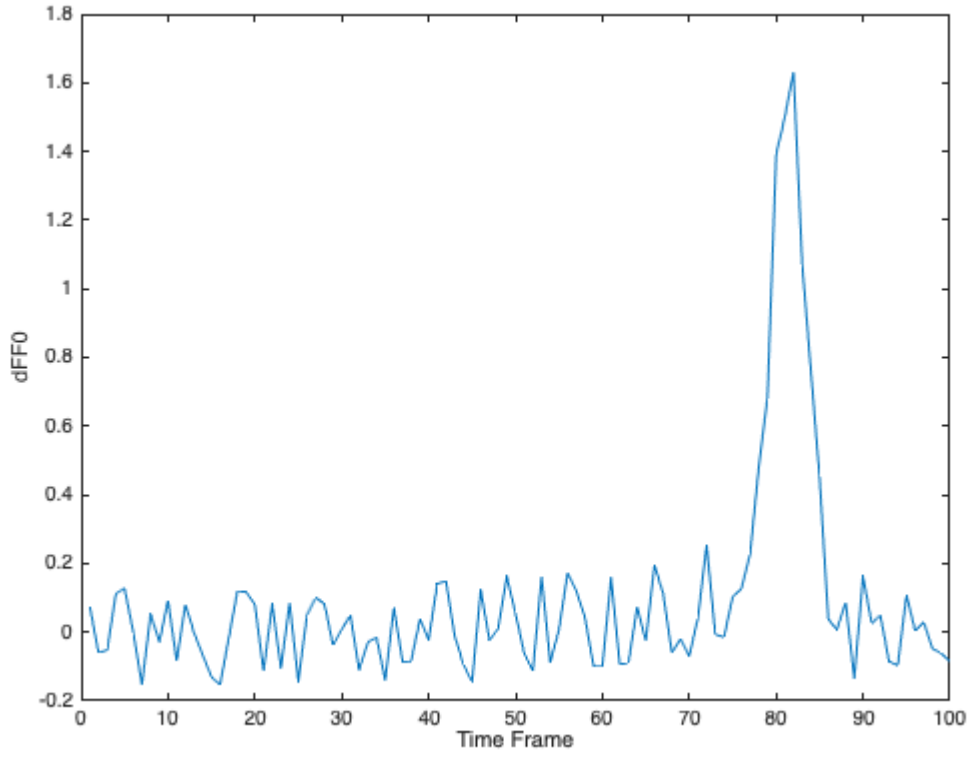
KyogoSKobayashi/Kobayashi_and_Matsuo_CellReports2023 is licensed under the MIT License

A short and simple permissive license with conditions only requiring preservation of copyright and license notices. Licensed works, modifications, and larger works may be distributed under different terms and without source code.

Generate DEMO data

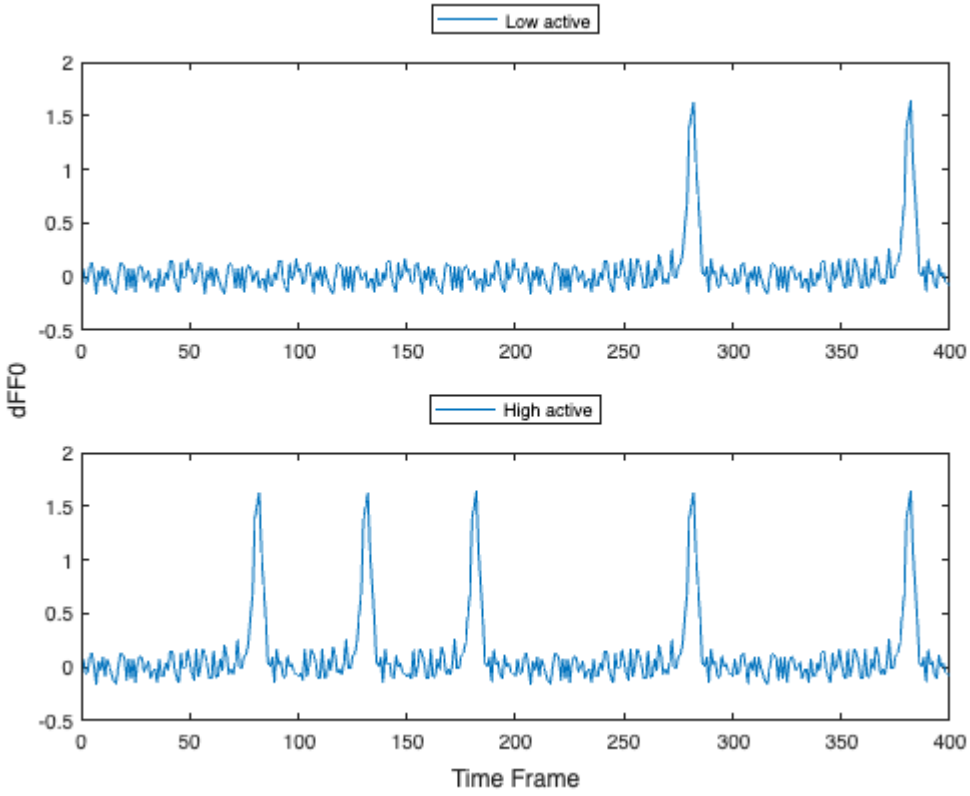
1. dFF0_data: 1600 frames-by-30 cells matrix of ΔF / F0 data .
2. Session_labels: 4-by-1 string vector; "ContextA1","ContextA2","ContextB1", and "ContextB2".
3. Frame_idx: 1600-by-1 string vector; "ContextA1","ContextA2","ContextB1", or "ContextB2".

```
recording_Hz = 20;

num_of_recording_frames_of_a_session = 400; % 20/400 = 20 sec
num_of_recording_cells = 30;

dFF0_data=zeros(num_of_recording_frames_of_a_session*4,num_of_recording_cells);
Session_labels = ["ContextA1";"ContextA2";"ContextB1";"ContextB2"];
Frame_idx = [repmat(Session_labels(1),num_of_recording_frames_of_a_session,1);...
             repmat(Session_labels(2),num_of_recording_frames_of_a_session,1);...
             repmat(Session_labels(3),num_of_recording_frames_of_a_session,1);...
             repmat(Session_labels(4),num_of_recording_frames_of_a_session,1)];


[~, y0] = titanium; %1-by-49 vector

rng default % for reproducibily
y1 = [y0,0.6]; %1-by-50 vector
y2 = y1-min(y1); %1-by-50 vector, represents a calcium event
one_event_trace = y2+(0.5-rand(size(y2)))/3; %1-by-50 vector with noise
no_event_trace = (0.5-rand(size(y2)))/3; %1-by-50 vector, bakcground fluctuation

figure; plot([no_event_trace,one_event_trace]);
ylabel ("dFF0");
xlabel("Time Frame");
```



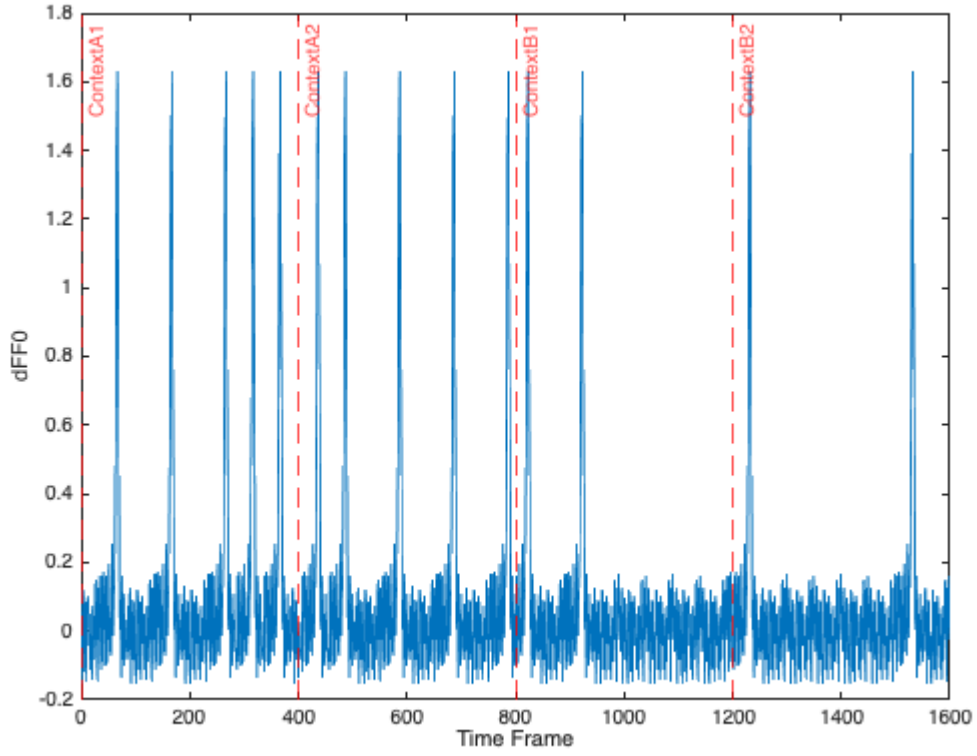

```
% high activity data
high_activity_data = [];
for ff = 1:(num_of_recording_frames_of_a_session/numel(y2))
    r = rand(1);
    if r > 0.3
        y = one_event_trace;
    else
        y = no_event_trace;
    end
    high_activity_data = [high_activity_data, y];
end

% low activity data
low_activity_data = [];
for ff = 1:(num_of_recording_frames_of_a_session/numel(y2))
    r = rand(1);
    if r > 0.8
        y = one_event_trace;
    else
        y = no_event_trace;
    end
    low_activity_data = [low_activity_data, y];
end

figure; t = tiledlayout("flow");
```

```
nexttile
plot(low_activity_data); legend("Low active",Location="northoutside")
nexttile
plot(high_activity_data); legend("High active",Location="northoutside")
ylabel (t, "dFF0");
xlabel(t, "Time Frame");
```




```
f = Frame_idx == "ContextA1";
d = high_activity_data;
for cc = 1:num_of_recording_cells
    r = randperm(num_of_recording_frames_of_a_session,1);
    dFF0_data(f,cc) = [d(r:end),d(1:r-1)];
end

f = Frame_idx == "ContextA2";
d = high_activity_data;
for cc = 1:num_of_recording_cells
    r = randperm(num_of_recording_frames_of_a_session,1);
    dFF0_data(f,cc) = [d(r:end),d(1:r-1)];
end

f = Frame_idx == "ContextB1";
d = low_activity_data;
for cc = 1:num_of_recording_cells
    r = randperm(num_of_recording_frames_of_a_session,1);
    dFF0_data(f,cc) = [d(r:end),d(1:r-1)];
```

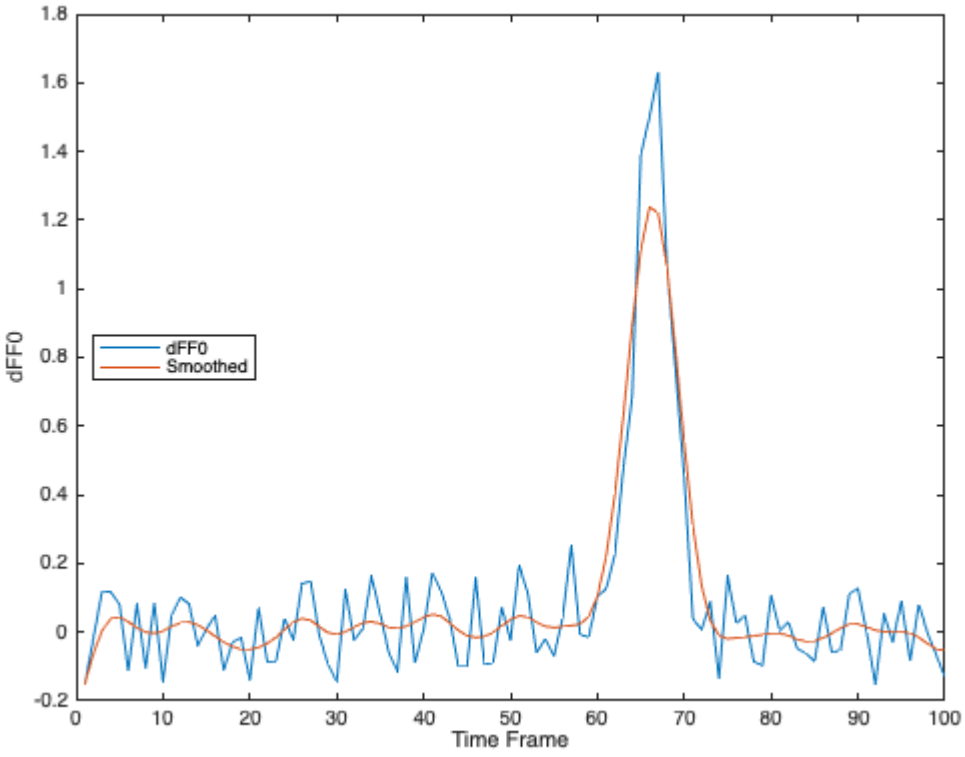
```
end

f = Frame_idx == "ContextB2";
d = low_activity_data;
for cc = 1:num_of_recording_cells
    r = randperm(num_of_recording_frames_of_a_session,1);
    dFF0_data(f,cc) = [d(r:end),d(1:r-1)];
end

figure;
plot(dFF0_data(:,1));
xline(1,'--r',Session_labels(1));
xline(num_of_recording_frames_of_a_session+1,'--r',Session_labels(2));
xline(num_of_recording_frames_of_a_session*2+1,'--r',Session_labels(3));
xline(num_of_recording_frames_of_a_session*3+1,'--r',Session_labels(4));
ylabel ("dFF0");
xlabel("Time Frame");
```



Calcium Event Detetion

1. Smoothed_dFF0_data: 1600 frames-by-30 cells matrix
2. Normalized_dFF0_data: 1600 frames-by-30 cells matrix
3. First_diff_dFF0_data: 1600 frames-by-30 cells matrix
4. Cal_Eve_Binary_data: 1600 frames-by-30 cells matrix

```
z_threshold = 2;

Smoothed_dFF0_data = dFF0_data*0;
[bFilt,aFilt] = butter(2,  2/(recording_Hz/2), 'low');
for ss = 1:numel(Session_labels)
    d1 = dFF0_data(Frame_idx == Session_labels(ss),:);
    smooted_d1 = filtfilt(bFilt,aFilt,d1);
    Smoothed_dFF0_data(Frame_idx == Session_labels(ss),:) = smooted_d1;
end

figure;
plot (dFF0_data(1:100,1)); hold on
plot (Smoothed_dFF0_data(1:100,1)); legend(["dFF0","Smoothed"],Location="west");
ylabel ("dFF0");
xlabel("Time Frame");
```




```
Normalized_dFF0_data = dFF0_data*0;
for ss = 1:numel(Session_labels)
    d1 = Smoothed_dFF0_data(Frame_idx == Session_labels(ss),:);
    d1_sd = std(d1,0,1);
    normalized_d1 = d1./d1_sd;
    Normalized_dFF0_data(Frame_idx == Session_labels(ss),:) = normalized_d1;
end

First_diff_dFF0_data = dFF0_data*0;
```

```
for ss = 1:numel(Session_labels)
    d1 = Normalized_dFF0_data(Frame_idx == Session_labels(ss),:);
    d1_diff = diff(d1);
    zeros_d1_diff = [zeros(1,num_of_recording_cells);d1_diff];
    First_diff_dFF0_data(Frame_idx == Session_labels(ss),:) = zeros_d1_diff;
end

Cal_Eve_Binary_data = dFF0_data*0;
ones_data = ones(size(dFF0_data));
binary_threshold = Normalized_dFF0_data > z_threshold & First_diff_dFF0_data > 0;
Cal_Eve_Binary_data(binary_threshold) = ones_data(binary_threshold);


figure;
plot (Normalized_dFF0_data(1:100,1)); hold on
plot (First_diff_dFF0_data(1:100,1));
yline(z_threshold,'--r',"2SD");
stem(Cal_Eve_Binary_data(1:100,1));
legend(["Normalized","1st diff","2SD","calcium events"],Location="northwest");
ylabel ("Normalized dFF0");
xlabel("Time Frame");
```